Computer Architecture - HW 1

Connor Finley

September 2, 2017

1

Consider three different processors P1, P2 and P3 executing the same instruction set. - P1 has a 3 GHz clock rate and a CPI of 1.5. - P2 has a 2.5 GHz clock rate and a CPI of 1.0. - P3 has a 4.0 GHz clock rate and has a CPI of 2.2.

a.

Which processor has the highest performance expressed in **instructions per second**?

1 GHz = 10ˆ9 cycles/sec

3 GHz = 3 * 10ˆ9 cycles/sec

1.5 CPU cycles / instruction count

$$\frac{3\times 10^9 \text{ cycles}}{1 \text{ second}}, \quad \frac{1.5 \text{ cycles}}{1 \text{ instruction}}, \quad \frac{?\ \text{instructions}}{1 \text{ second}}$$

$$\text{P1 :} \quad \frac{3\times 10^9 \text{ cycles}}{1 \text{ second}} \div \frac{1 \text{ instruction}}{1.5 \text{ cycles}}$$

$$= \frac{3\times 10^9 \text{ instructions}}{1.5 \text{ seconds}} = 2\times 10^9 \text{ instructions per second}$$

$$\text{P2 :} \quad \frac{2.5\times 10^9 \text{ cycles}}{1 \text{ second}} \div \frac{1 \text{ instruction}}{1 \text{ cycles}}$$

$$= 2.5\times 10^9 \text{ instructions per second}$$

$$\text{P3 :} \quad \frac{4\times 10^9 \text{ cycles}}{1 \text{ second}} \div \frac{1 \text{ instruction}}{2.2 \text{ cycles}}$$

$$= 1.82\times 10^9 \text{ instructions per second}$$

$$P3 = 1.82\times 10^9 < P1 = 2\times 10^9 < P2 = 2.5\times 10^9$$

P2 has the highest performance with 2.5×10^9 instructions per second.

b.

If the processors each execute a program in 10 seconds, find the number of cycles and the number of instructions.

$$P1 = 10 \text{ seconds} \times 3 \text{ GHz} = 10 \text{ seconds} \times (3 \times 10^9 \text{ cycles per second}) = 3 \times 10^{10} \text{ cycles}$$

$$\frac{3 \times 10^{10} \text{ instructions}}{1.5 \text{ seconds}} = 2 \times 10^{10} \text{ instructions per second}$$

$$10 \text{ seconds} \times (2 \times 10^{10} \text{ instructions per second}) = 2 \times 10^{11} \text{ instructions}$$

$$P2 = 10 \text{ seconds} \times 2.5 \text{ GHz} = 10 \text{ seconds} \times (2.5 \times 10^9 \text{ cycles per second}) = 2.5 \times 10^{10} \text{ cycles}$$

$$\frac{2.5 \times 10^{10} \text{ instructions}}{1 \text{ seconds}} = 2.5 \times 10^{10} \text{ instructions per second}$$

$$10 \text{ seconds} \times (2.5 \times 10^{10} \text{ instructions per second}) = 2.5 \times 10^{11} \text{ instructions}$$

$$P3 = 10 \text{ seconds} \times 4 \text{ GHz} = 10 \text{ seconds} \times (4 \times 10^9 \text{ cycles per second}) = 4 \times 10^{10} \text{ cycles}$$

$$\frac{4 \times 10^{10} \text{ instructions}}{2.2 \text{ seconds}} = 1.82 \times 10^{10} \text{ instructions per second}$$

$$10 \text{ seconds} \times (1.82 \times 10^{10} \text{ instructions per second}) = 1.82 \times 10^{11} \text{ instructions}$$

---

c.

We are trying to execute the same program as in part b), and we are trying to reduce the execution time by 30% but this leads to an increase of 20% in the CPI. What clock rate should we have to get this time reduction?

- Target: 70% of old execution time (10 seconds), or 7 seconds

CPI of **P1** = 1.5, increase of 20% = 1.5 × 1.2 = 1.8

$$\frac{2 \times 10^{11} \text{ instructions}}{7 \text{ seconds}} \times \frac{1.8 \text{ cycles}}{1 \text{ instruction}} = 0.51 \times 10^{11} \text{ cycles per second} = 51 \text{ GHz}$$

CPI of **P2** = 1, increase of 20% = 1 × 1.2 = 1.2

$$\frac{2.5 \times 10^{11} \text{ instructions}}{7 \text{ seconds}} \times \frac{1.2 \text{ cycles}}{1 \text{ instruction}} = 0.43 \times 10^{11} \text{ cycles per second} = 43 \text{ GHz}$$

CPI of **P3** $= 2.2$, increase of $20\% = 2.2 \times 1.2 = 2.64$

$$\frac{1.82 \times 10^{11} \text{ instructions}}{7 \text{ seconds}} \times \frac{2.64 \text{ cycles}}{1 \text{ instruction}} = 0.69 \times 10^{11} \text{ cycles per second} = 69 \text{ GHz}$$

2.

Consider two different implementations of the same instruction set architecture. The instructions can be divided into four classes according to their CPI (class A, B, C and D). P1 with a clock rate of 2.5 GHz and CPIs of 1, 2, 3 and 3, and P2 with a clock rate of 3 GHz and CPIs of 2, 2, 2 and 2.

Given a program with a dynamic instruction count of 1.0E6 instructions divided into classes as follows: 10% class A, 20% class B, 50% class C and 20% class D, which implementation is faster?

Instruction class	P1 (2.5 GHz) CPI	P2 (3 GHz) CPI	Instruction count
A	1	2	100,000
B	2	2	200,000
C	3	2	500,000
D	3	2	200,000

P1: 2.5×10^9 cycles/sec

P1 (A): 1 cycle per instructions with 100,000 instructions = 100,000 cycles $=> 100,000 \div \frac{2.5 \times 10^9 \text{ cycles}}{1 \text{ second}} = 0.00004$ seconds

P1 (B): 2 cycles per instruction with 200,000 instructions = 400,000 cycles $=> 400,000 \div \frac{2.5 \times 10^9 \text{ cycles}}{1 second} = 0.00016$ seconds

P1 (C): 3 cycles per instruction with 500,000 instructions = 1.5 million cycles $=> 1,500,000 \div \frac{2.5 \times 10^9 \text{ cycles}}{1 \text{ second}} =$ 0.0006 seconds

P1 (D): 3 cycles per instruction with 200,000 instructions = 600,000 cycles $=> 600,000 \div \frac{2.5 \times 10^9 \text{ cycles}}{1 \text{ second}} = 0.00024$ seconds

P1 total $= 0.00004 + 0.00016 + 0.0006 + 0.00024 = 0.00104$ seconds

---

P2: 3×10^9 cycles/sec

P2 (A): 2 cycles per instructions with 100,000 instructions = 200,000 cycles $=> 200,000 \div \frac{2.5 \times 10^9 \text{ cycles}}{1 \text{ second}} =$ 0.00008 seconds

P2 (B): 2 cycles per instruction with 200,000 instructions = 400,000 cycles $=> 400,000 \div \frac{2.5 \times 10^9 \text{ cycles}}{1 \text{ second}} = 0.00016$ seconds

P2 (C): 2 cycles per instruction with 500,000 instructions = 1 million cycles $=> 1,000,000 \div \frac{2.5 \times 10^9 \text{ cycles}}{1 \text{ second}} =$ 0.0004 seconds

P2 (D): 2 cycles per instruction with 200,000 instructions = 400,000 cycles $=> 600,000 \div \frac{2.5 \times 10^9 \text{ cycles}}{1 \text{ second}} = 0.00016$ seconds

P2 total $= 0.00008 + 0.00016 + 0.0004 + 0.00016 = 0.0008$ seconds

P2 is a faster implementation

a.

What is the global CPI for each implementation?

P1: $(1 \times 0.1) + (2 \times 0.2) + (3 \times 0.5) + (3 \times 0.2) = 0.1 + 0.4 + 1.5 + 0.6 = 2.6$

P2: $(2 \times 0.1) + (2 \times 0.2) + (2 \times 0.5) + (2 \times 0.2) = 0.2 + 0.4 + 1 + 0.4 = 2$

b.

Find the clock cycles required in both cases.

P1: $1,000,000$ instructions $\times \frac{2.6 \text{ cycles}}{1 \text{ instruction}} = 2,600,000$ clock cycles

P2: $1,000,000$ instructions $\times \frac{2 \text{ cycles}}{1 \text{ instruction}} = 2,000,000$ clock cycles

3

Assume for arithmetic, load/store, and branch instructions, a processor has CPIs of 1, 12, and 5, respectively. Also assume that on a single processor, a program requires the execution of 2.56E9 arithmetic instructions, 1.28E9 load/store instructions, and 256 million branch instructions. Assume that each processor has a 2 GHz clock frequency.

Assume that, as the program is parallelized to run over multiple cores, the number of arithmetic and load/store instructions per processor is divided by (0.7 * p) (where p is the number of processors) but the number of branch instructions per processor remains the same.

Instruction Type	Instruction CPI	Instruction Count
Arithmetic	1	2,560,000,000
Load/Store	12	1,280,000,000
Branch	5	256,000,000

a.

Find the total execution time for this program on 1, 2, 4 and 8 processors, and show the relative speedup of the 2, 4, and 8 processor result relative to the single processor result.

1 processor

- Arithmetic
  - 1 CPI * 2.6 billion instructions = 2.6 billion cycles => 2.6 billion cycles / 2 GHz = 1.3 seconds
- Load/Store
  - 12 CPI * 1.28 billion instructions = 15.36 billion cycles => 15.36 billion cycles / 2 GHz = 7.68 seconds
- Branch
  - 5 CPI * 256 million instructions = 1.28 billion cycles => 1.28 billion cycles / 2 GHz = 0.64 seconds
- Total execution time: 1.3 seconds + 7.68 seconds + 0.64 seconds = 9.62 seconds

2 processors

- Arithmetic
  - 1 CPI * (2.6 billion instructions / (0.7 * 2)) = 1,857,142,857 cycles => 0.928 seconds
  - Speedup = 1.3 seconds / 0.928 seconds = 1.4X

- Load/Store
  - 12 CPI * (1.28 billion instructions / (0.7 * 2)) = 10,971,428,571 cycles => 5.485 seconds
  - Speedup = 7.68 seconds / 5.485 seconds = 1.4X
- Branch
  - No change (0.64 seconds)
  - Speedup: 1X
- Total execution time: 0.928 seconds + 5.485 seconds + 0.64 seconds = 7.053 seconds

4 processors

- Arithmetic
  - 1 CPI * (2.6B instructions / (0.7 * 4)) = 0.92B cycles => 0.46 seconds
  - Speedup = 1.3 seconds / 0.46 seconds = 2.83X
- Load/Store
  - 12 CPI * (1.28B instructions / (0.7 * 4)) = 5.485B cycles => 2.74 seconds
  - Speedup = 7.68 seconds / 2.74 seconds = 2.8X
- Branch
  - No change (0.64 seconds)
  - Speedup: 1X
- Total execution time: 0.46 seconds + 2.74 seconds + 0.64 seconds = 3.84 seconds

8 processors

- Arithmetic
  - 1 CPI * (2.6B instructions / (0.7 * 8)) = 0.46B cycles => 0.23 seconds
  - Speedup = 1.3 seconds / 0.23 seconds = 5.65X
- Load/Store
  - 12 CPI * (1.28B instructions / (0.7 * 8)) = 2.74B cycles => 1.37 seconds
  - Speedup = 7.68 seconds / 1.37 seconds = 5.6X
- Branch
  - No change (0.64 seconds)
  - Speedup: 1X
- Total execution time: 0.23 seconds + 1.37 seconds + 0.64 seconds = 2.24 seconds

b.

> If the CPI of the arithmetic instructions was doubled, what would the impact be on the execution time of the program on 1, 2, 4 or 8 processors?

The number of cycles and execution time for the arithmetic portion would double.

- 1 processor: Total execution time: (1.3 seconds * 2) + 7.68 seconds + 0.64 seconds = 10.92 seconds
- 2 processors: Total execution time: (0.928 seconds * 2) + 5.485 seconds + 0.64 seconds = 7.981 seconds
- 4 processors: Total execution time: (0.46 seconds * 2) + 2.74 seconds + 0.64 seconds = 4.3 seconds
- 8 processors: Total execution time: (0.23 seconds * 2) + 1.37 seconds + 0.64 seconds = 2.47 seconds

c.

> To what should the CPI of load/store instruction be reduced in order for a single processor to match the performance of four processors using the original CPI values.

Total cycle count with 4 processors: 7.685B cycles

1 processor: (1 CPI * 2.6 billion instructions) + (x CPI * 1.28 billion instructions) + (5 CPI * 256 million instructions)

7.685B cycles = 1 processor

7.685B cycles = (1 CPI * 2.6 billion instructions) + (x CPI * 1.28 billion instructions) + (5 CPI * 256 million instructions)

7.685B cycles = 2.6B cycles + (1.28B instructions)(x CPI) cycles + 1.28B cycles

3.805B cycles = 1.28B * x

x = 2.973 CPI

Check: 2.973 CPI * 1.28B instructions = 3,805,440,000 cycles =>3,805,440,000 cycles / 2 GHz = 1.90272 seconds

1.90272 seconds + 1.3 seconds + 0.64 seconds = 3.84 seconds = total execution time for 4 processors

4.

Consider a computer running a program that requires 250 s, with 70 s spent executing float-point (FP) instructions, 85 s executed load/store (L/S) instructions, and 40 s spent executing branch instructions and 55 s spent executing INT instructions.

a.

By how much is the total time reduced if the time for FP operations is reduced by 20%?

70 s * .8 = 59.2 s => total time - 10.797 s = 250 - 10.797 = 239.2 seconds

b.

By how much is the time for INT operations reduced if the total time is reduced by 20%?

Going off of initial percentages, FP = 28%, L/S = 34%, branch = 16%, INT = 22%.

250 - 20% = 200, 22% of 200 = 44 seconds. 55 -44 = 11 second reduction

c.

Can the total time be reduced by 20% by reducing only the time for branch instructions?

250 -20% = 200, delta = 50 seconds

Total branch time: 40s

40 seconds < 50 seconds so, therefore, no. This time reduction can't be reached, even by eliminating branch instructions completely.

5.

Assume that we are considering enhancing a machine by adding vector hardware to it. When a computation is run in vector mode on the vector hardware, it is 10 times faster than the normal mode of execution. We call the percentage of time that could be spent using vector mode the percentage of vectorization.

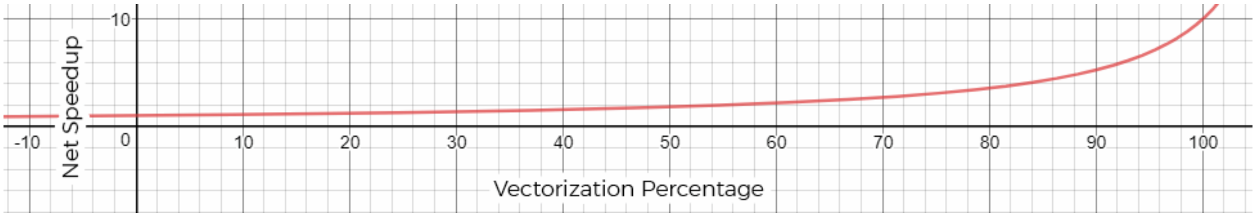

Figure 1:

a.

Draw a graph that plots the speedup as a percentage of the computation performed in vector mode. Label the y-axis “Net Speedup” and label x-axis “Percentage Vectorization.”

b.

What percentage of vectorization is need to achieve a speedup of 2?

S = speedup overall, F = vectorized fraction, P = vectorized speedup

$$S = \frac{1}{(1-F)+\frac{F}{P}}$$

$S = 2,$

$F = ?,$

$P = 10$

$$2 = \frac{1}{(1-F)+\frac{F}{10}}$$

$$2 = \frac{1}{1-0.9F}$$

$$2(1-0.9F) = 1$$

$$1 = 1.8F$$

$$F = 0.55 \;\; or \;\; 55.55\%$$

c.

What percentage of computation run time is spent in vector mode if a speedup of 2 is achieved?

Speedup of 2 means 55.55% vectorization, 44.4% normal (original total time - F).

Speedup of 2 also means the original total time is halved.

halved total time - normal time = vectorized time.

Let original total time = 1000 seconds.

44.4% of 1000 = 444 seconds

Halved time = 500 seconds.

500 - 444 = 56 seconds

56 / 1000 seconds = **5.6% of time in vector mode**

d.

> What percentage of vectorization is needed to achieve one-half the maximum speedup attainable from using vector model?

Half the max speedup = 5.

$$5 = \frac{1}{1 - 0.9F}$$

$$5(1 - 0.9F) = 1$$

$$5 - 4.5F = 1$$

$$4 = 4.5F => F = 0.8888 = 88.89\%$$

6.

> Assume that we make an enhancement to a computer that improves some mode of execution by a factor of 10. Enhanced mode is used 50% of the time, measure as a percentage of the execution time when the enhanced mode is in use. Recall that Amdahl's law depends on the fraction of the original, unenhanced execution time that could make use of enhanced mode. Thus, we cannot directly use this 50% measurement to compute speedup with Amdahl's law.

a.

> What is the speed up we have obtained from fast mode?

50% of time in normal mode, 50% of time in enhanced mode.

Enhanced mode is 10 times faster than normal mode, so, to get the same time fraction as normal mode, enhanced mode is actually 500% in normal mode, but sped up. Using this, the total time is 550% in normal mode.

Using Amdahl's Law, speedup = exec. time of old / exec. time of new

speedup = 550% / 100% = 5.5

b.

> What percentage of the original execution time has been converted to fast mode?

S = speedup overall, F = vectorized fraction, P = vectorized speedup

$$S = \frac{1}{(1-F) + \frac{F}{P}}$$

$S = 5.5,$

$F = ?,$

$P = 10$

$$5.5 = \frac{1}{(1-F) + \frac{F}{10}}$$

$$5.5 = \frac{1}{1 - 0.9F}$$

$$5.5(1 - 0.9F) = 1$$

$$5.5 - 4.95F = 1$$

$$4.5 = 4.95F => F = 0.90909 = 90.9\%$$